A Modular Access Control Architecture for the Earth System Grid Federation

Philip Kershaw (philip.kershaw@stfc.ac.uk) (1), Rachana Ananthakrishnan (ranantha@mcs.anl.gov) (2), Luca Cinquini (Luca.Cinquini@jpl.nasa.gov) (3) (4), Dennis Heimbigner (dmh@unidata.ucar.edu) (5), and Bryan Lawrence (bryan.lawrence@stfc.ac.uk) (1)

(1) STFC Rutherford Appleton Laboratory, NCAS/British Atmospheric Data Centre, Didcot, United Kingdom, (2) Argonne National Laboratory, Argonne, IL, USA, (3) Jet Propulsion Laboratory, National Aeronautics and Space Administration, Pasadena, CA, USA, (4) Earth System Research Laboratory, National Oceanic and Atmospheric Administration, Boulder, CO, USA (5) University Corporation for Atmospheric Research, Boulder, CO, USA

For submission to GCA 2011

Keywords: CMIP5, ESGF, OPeNDAP, NetCDF, security

Abstract

We present key aspects of the federated access control solution required for the Earth System Grid Federation (ESGF), including a standard mechanism for securing OPeNDAP (Open-source Project for a Network Data Access Protocol) based services and corresponding extensions to the NetCDF (Network Common Data Form) software libraries to support this paradigm.

ESGF is an international collaboration to facilitate and empower access to the analysis of earth science data – beginning with a deployment in support of the Coupled Model Intercomparison Project, Phase 5 (CMIP5). CMIP5 is a framework of climate model experiments, involving the production of and interpretation of data from several modelling centres around the world. The results from this activity will be available in a globally accessible archive mirrored at a number of key sites. While much of this data will be available with unrestricted access, the archives still need a security infrastructure to control demand, and ensure data integrity and report usage metrics.

Developing a security architecture for such a system presents significant challenges. This is highlighted by the heterogeneous nature of the environment in which a solution must be applied. This diversity is expressed on a number of levels: the range of tools and services used within the climate model community, the associated protocols and technology stacks employed, and the varied organisational structures and domains representative across the federation members. By maintaining a separation of concerns between the various aspects it has been possible to devise a highly flexible access control architecture adaptable to the spectrum of needs presented in the system. Such a modular approach is only possible through the definition of interfaces: at the inter-organisational level with web services and at the application level with the use of server side middleware and REST based principles.

1. Introduction

The production, evaluation and interpretation of climate model simulations are integral activities within earth system science. Since the very first "General Circulation Models" run more than forty years ago, to the very latest "Earth System Model" simulations running now as part of the fifth Coupled Model Intercomparison Project (CMIP5), these activities have always been on the leading edge of computing - requiring the largest computers and huge data archives. As the models have improved, adding more internal processes, and running at higher resolution, so has the volume of data produced increased.

CMIP5, organised under the auspices of the World Climate Research Programme (WCRP) will deliver science that will feed into the next Intergovernmental Panel on Climate Change (IPCC) assessment report. As such, the analysis and interpretation will be a global activity, requiring global access to petascale data archives held on multiple continents. Traditional centralised archive solutions will clearly not suffice.

However, just as clearly, the scientific community will not respond well to working with data held in disparate formats with a variety of access techniques. CMIP5 has itself developed a protocol which addresses data formats and data licensing, but it could not itself develop a data management solution. To that end, a global federation – the "Earth System Grid Federation" has been built on the nucleus of the U.S. Earth System Grid Center for Enabling Technologies (ESG-CET)[1]. The Earth System Grid Federation was established to cope with data production at o(25) sites globally conforming to o(50) distinct numerical experiments and resulting in o(100,000) years of simulated climate corresponding to o(6500) years of the real world climate.

A key tenet of the design philosophy of CMIP5 was to identify the "core" output from the simulations – that is the data which was likely to see the most analysis by scientists. The consequential key requirement for ESGF was and is to maximise the exposure of that core data (expected to be approximately 2.5 petabytes), even as it exposes all the data produced for CMIP5. ESGF then is essentially a federation of the originating modelling archives (or their proxies), and a number of *replicant* archives – three of which have committed to persist that core data indefinitely. These persistent archives will be located at the Program for Climate Model Diagnosis and Intercomparison at the U.S. Lawrence Livermore National Laboratory, the British Atmospheric Data Centre in the UK National Centre for Atmospheric Science, and the German Climate Computing Centre. ESGF itself will be described in detail elsewhere[22], the purpose of this paper is to describe particular techniques that have been applied to design the access control to the data in this globally distributed data system.

We first consider the security requirements, and then the modular architecture that has been designed and applied to address those requirements. It will be seen that a critical requirement was to ensure that familiar tools could integrate with the access control architecture with little or no modification. We focus on (1) HTTP based services and key applications (2) the NetCDF client implementation of the OPeNDAP protocol and (3) wget scripts. Finally, a walkthrough of a typical use case illustrates how the various components come together in the working system.

2. Requirements

ESGF has some key requirements that motivate the security solutions provided:

1) Seamless access to data hosted by all organizations in the federation, that is, single sign-on such that the same credentials can be used across the federation
2) A mechanism to set policy on restricting access to chosen datasets, per dataset on a case by case basis
3) The ability to notify users of changes to data and services. This requires the collection of user attributes including e-mail addresses whilst at the same time respecting user privacy.
4) The ability to collect metrics about data download, specifically the number of unique downloads
5) Seamless integration with multiple interfaces to a service or resource. Specifically, browser based access and thick client access.
6) Clean integration with services and tools that scientists commonly use.

These requirements are considered each in turn in the following sections, but we begin by outlining the overarching deployment environment and architectural requirements.

The ESGF architecture defines Data Nodes and Gateway Nodes. Data Nodes are sites that host the model data and associated access services. Replication services enable the CMIP5 core data to be mirrored across the key archiving sites and publishing services make that data discoverable through Gateways, portals to the system.

For requirement 1), we look at the application of a service oriented architecture, and requirements 2), 3) and 4) are addressed in Attribute Management and Authorization solutions. Considering 5), ESGF includes both GridFTP[14] and HTTP based data access services. However, for the purposes of this paper we concentrate on the HTTP server side architecture and how this has been implemented to maintain a separation of concerns between access control functionality and underlying applications to be secured. Finally 6) focusses on work carried out for ESGF to add a standardised access control layer to OPeNDAP[12], a core data access service for the federation.

Significantly for the security architecture, PCMDI has a lead role for CMIP5, holding the delegated authority of the various modelling groups to allow access according to their varying access criteria (co-ordinated by the WCRP). As such it needs to control the assignation of CMIP5 access authorisation, on a dataset by dataset basis, to individuals in the user community. Each ESGF institution may host CMIP5 datasets other than their own, but in doing so they need to honour the PCMDI role, even as they retain control over their own datasets, and those under other authorisation domains.

Much of the data will be available with liberal licensing conditions. ESGF currently expects that a simple registration hurdle, coupled with the requirement for an e-mail address that can be validated. Thus, the level of assurance required is low in comparison to many systems. That said, for resource providers, the security architecture should provide some level of protection for their finite computing assets, for example from malicious or unintended requests, which might overload network or server resources.

Finally, to function as a federation the ability to collect, curate, and publish trusted federation service metadata is key.

3. Service Oriented Architecture

ESGF is deployed in a variety of locations, alongside existing activities. Fundamental then to the development of a federated access control infrastructure is the interfaces between organisations. A standards based approach was employed wherever practicable to facilitate interoperability and ensure the use of peer reviewed protocols. In this section we describe the services and their interfaces: looking in turn at authentication and single sign-on, attribute management and authorisation.

3.1. Authentication and Single Sign-on

The distributed nature of the ESG architecture meant that single sign-on was favoured from the outset as a means to simplify access for users and join the user management infrastructures of the different participating institutions together.

The OpenID[9] standard was chosen early by the ESG team to provide single sign-on capability. An evaluation exercise showed that particular vulnerabilities in the specification could be addressed by stipulating SSL for OpenID Provider endpoints. As a consequence, ESGF OpenID Relying Parties are able to utilise SSL based peer authentication to whitelist OpenID Provider identities to a given set of registered Identify Providers (IdPs) within the federation. The restricted set of IdPs allowed ESGF to leverage an agreed set of site attributes, and enforce trust and Service Level Agreements (SLA) on the IdP. Each ESGF Gateway Node hosts an OpenID Identity Provider, where a user can register to get a login.

OpenID is augmented with the use of SAML[8] (the Security Assertion Mark-up Language) v2.0 with the SOAP (Simple Object Access Protocol) binding to provide standard interfaces for the various other security services required to broker access. As a baseline, all interactions with services are secured with Transport Layer Security (TLS), with mutual authentication. Again, whitelisting of client certificate subject names enables services to restrict queries to a trusted set of retrievers.

Dual Authentication Mechanisms
While OpenID is suited for interaction with browser clients, it does not lend itself well to use with thick clients. To support the latter, a credential translation system that converted the OpenID token to a token format consumable by the thick clients was required. The chosen token format for such use cases was X.509 Certificates[17]. Thus in addition to the OpenID Identity Provider each Gateway Node site runs a MyProxy[11] Online CA service, that can issues a short term X.509 Credential which can be used with PKI (Public Key Infrastructure) aware applications. The Online CA is backed by the same user authentication system as the OpenID service, thus issuing a certificate to any user who has a valid OpenID login. Short-lived credentials issued from the MyProxy server are configured to include the given OpenID URI in the certificate subject name. This mechanism is used for authentication with GridFTP[14] servers and adapted for HTTP based applications including OPeNDAP services as will be described later in this paper.

3.2. Attribute Management

User attributes are exchanged between trusted parties within the federation. They fall into two categories which derive directly from items 2) and 3) listed in the requirements section:

- Site attributes include a limited amount of personal user information used for registration and notification purposes and are specific to the user's IdP, whereas,
- VO attributes include access control attributes used to restrict access to data and computational resources. They are scoped for the community and may be assigned at some other ESGF authority than their IdP via a registration process.

VO level attribute agreements were necessitated for two key use cases: access to the distributed CMIP5 data archive and bulk replication of data between archiving sites. For CMIP5 data access, PCMDI has authority to issue users with access rights. For the replication use case, the originating site of the data to be replicated has authority. In all cases, attributes names are namespace constrained to ensure enforcement of the issuing authority.

Push and Pull Models for Attribute Retrieval
Exploring these use cases it became apparent that the system would benefit from both push and pull models for the transmission of attributes to consumers.

Attributes may be pushed at the authentication stage: as a starting point, OpenID's AX (Attribute Exchange) mechanism was utilised. Analogous to this, prior work at ANL[5] had shown how MyProxy may be configured to embed attributes in X.509 certificate extensions as SAML assertions. This was applied for the replication use case where the authorisation layer of the GridFTP service, can extract the attribute assertion to determine access for a given resource.

In some scenarios such a pull model is more suited: where attribute information is required out of band of the authentication process or where the source of authority for attribute information is not itself an IdP,. Any authority such as PCMDI, which has responsibilities for a specific data set or a group of data sets, may host a SAML based *Attribute Service*. This service assigns attributes to the users for data access, and also provides an interface for consumers to query user attribute entitlement. These services are associated with the resources they protect, and may have users registered with them from a number of different IdPs from within the federation. Attribute services may also be deployed in association with an IdP should it for any reason be unable to support the push based approach for attribute propagation. Attribute Services employ whitelist techniques, based on the federation trusted service metadata (see following 3.4), to restrict access to user attributes and preserve user privacy.

3.3. Authorisation Service

Each organisation within ESGF that hosts secure services (e.g. OPeNDAP or GridFTP), also hosts an Authorisation Service which exposes a SAML interface which allows authorized remote entities in the ESGF to query for decisions on access to given resources. This service uses a pull model to obtain user attributes. A registry maps user attribute names onto their respective issuing Attribute Service, so that for example, a resource secured with a CMIP5 attribute will trigger a query to the PCMDI Attribute Service to verify the user's entitlement to this attribute.

3.4. Federation Metadata

An essential aspect of any federation is the establishment and curation of federation credentials, which provides the core trust roots of the federation. In the case of ESGF, for authentication purposes the following metadata is required:

1) Trusted CA certificates, Signing Policy, and CRLs (Certificate Revocation Lists), and
2) The whitelist of OpenID Identity Providers

1) is used to validate any certificate chain presented in on a TLS channel (both for client to service, and service-to-service communication). 2) is used by OpenID Relying Parties to restrict which IdPs can assert user identities in the federation.

In addition, the metadata also contains information about the various trusted services in the federation, including Attribute and Authorization services, and data download services, which may query them. We have defined a schema to describe the data, and are in the process of building an infrastructure that will allow each organization to own and register their metadata, and obtain the complete federation data for their use.

Service Discovery
OpenID 2.0, supports the Yadis[10] protocol whereby a HTTP GET request for a user's OpenID yields an XRDS document containing the service endpoint for the respective OpenID Provider. XRDS can be further exploited to advertise multiple identity services and in this case link them to a user's identity URL. Thus, a given OpenID may be introspected to discover identity services associated with that user's IdP. For ESGF, this has been leveraged to include the MyProxy server and Attribute Service

associated with the IdP.

4. Modular Architecture for HTTP Based Services

In this section, we describe the architecture adopted for integrating security with the HTTP based access services, without any changes to the core services themselves. Prior to the work with ESGF, lessons drawn from previous software development projects at the BADC[3] had highlighted the need for non-intrusive approaches to access control of HTTP services: to support the layering of access control functionality over existing services in such a way as to minimise the impact on their interfaces. Two strong themes emerged: the use of REST[5] based principles to govern access control policy and the use of Aspect Oriented Programming (AOP)[7] techniques.

Security is often cited as an exemplar for AOP. HTTP server-side interface specifications like the Python WSGI[15] (Web Server Gateway Interface) and Java Servlets provide a means to layer access control middleware components without the need to modify the underlying application. This is an important principle, as any alternative custom API for applying access control callouts would necessitate changes to the application code itself and thus the universal applicability of a generic solution would be lost.

This separation of concerns between access control functionality and the underlying application that it protects has further implications. The use of a given middleware interface specification constrains the range of properties upon which access may be determined to within the scope of the parameters of that interface: the HTTP request URI, method and so on. Adopting REST based principles, URIs may be associated with resources to be protected and so a URI based access control policy can be realised. This has the advantage of performance – request content need not be parsed, only the request URI – and clarity: resources to be protected have a clear mapping to the URIs by which they are exposed. Not all services are easily amenable to this practice however. For example, some operations for OGC (Open Geospatial Consortium) W*S require the use of the POST method. In such cases the access control middleware may need to consume the request message body so as to apply a given access policy.

A consequence of a URI based access policy is that the granularity of the URI scheme must match the granularity of access control policy required. In practice this has meant some careful consideration of the ESGF URI schemes for protected applications and data.

This whole approach lays a distinction from some security application frameworks where access control functionality is embedded in the application code itself. Whilst they provide flexibility and fine-grained control over access, they break the separation between application code and access control functionality. In general they cannot be deployed in environments where service stacks are maintained and developed independently of the security framework.

A filter-based architecture also enables the assembly of independent middleware components into a pipeline or chain since they all adhere to a common interface. This characteristic can be exploited to divide up access control functionality. For example, HTTP response codes can be utilised to separate the function of catching an unauthenticated request – by setting a HTTP 401 Unauthorized code - from the function of enforcing some associated response - e.g. displaying a sign in user interface.

Filter Chain for ESGF Services

Filters are defined to perform specific authentication and authorization related functions and follow a specific order. This is illustrated in figure below:

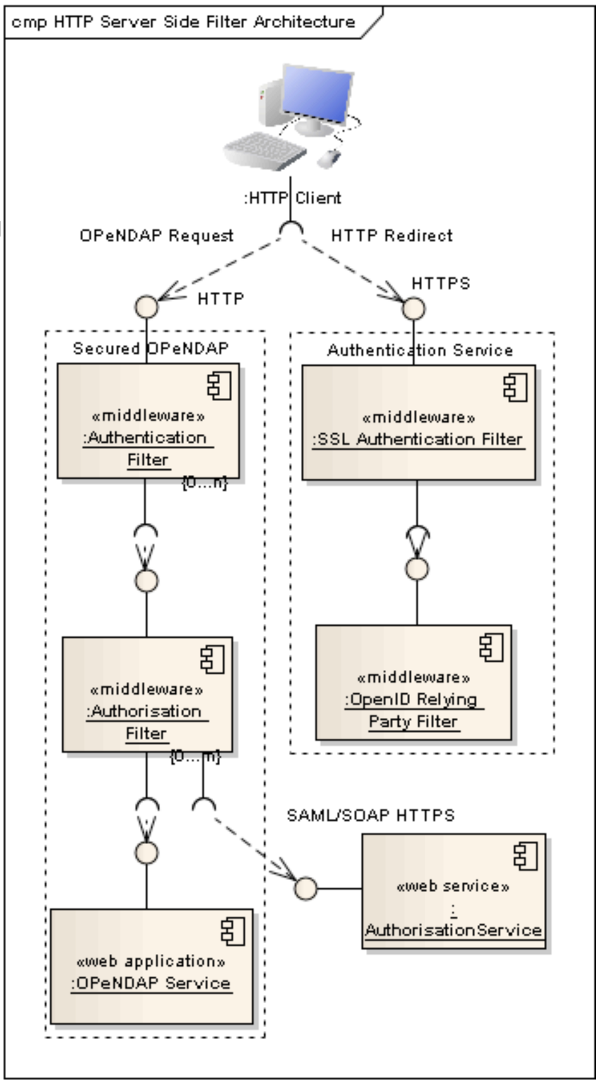

Figure 1: HTTP Server Side Filter Chain

Two filter chains are shown. The first fronts the application to be protected; the second one shown alongside it, deals specifically with the authentication process. The request from a client goes to the data serving application to be accessed, in this case an OPeNDAP service. An authentication filter is first to intercept the request. This checks the access restrictions for requested resource by consulting the policy. If no restriction is in place, control is passed on to the underlying application to serve the request. If a secured resource has been requested, the filter checks for the presence of a valid session cookie. In the absence of this, the client is returned a HTTP 30x response requesting redirection to an authentication service endpoint, which listens over HTTPS.

The authentication service uses a twofold chain to enable a client to authenticate with either PKI based credentials or via OpenID. Significantly, with this arrangement, the server side is agnostic to the client request method employed. The first filter checks for a user X.509 certificate obtained from the SSL handshake. If present, authentication proceeds based on verification of this identity, otherwise control passes to the next filter which initiates an OpenID Relying Party interface. The default behaviour then is to assume OpenID based sign in from a browser but note that the response code will be HTTP 401 Unauthorized to signal to non-browser based clients that authentication credentials are required.

Whatever authentication method is used, a positive result will trigger a HTTP 30x redirect response to return the client back to the HTTP based authentication filter. A signed authentication cookie is returned with this in the HTTP header. The recipient must be within the same cookie domain so that the returned cookie is visible to the authentication filter fronting the data serving application. On receipt of the cookie, this filter verifies it, sets the users authenticated status and passes control on to the next filter. The sequence below illustrates the steps:

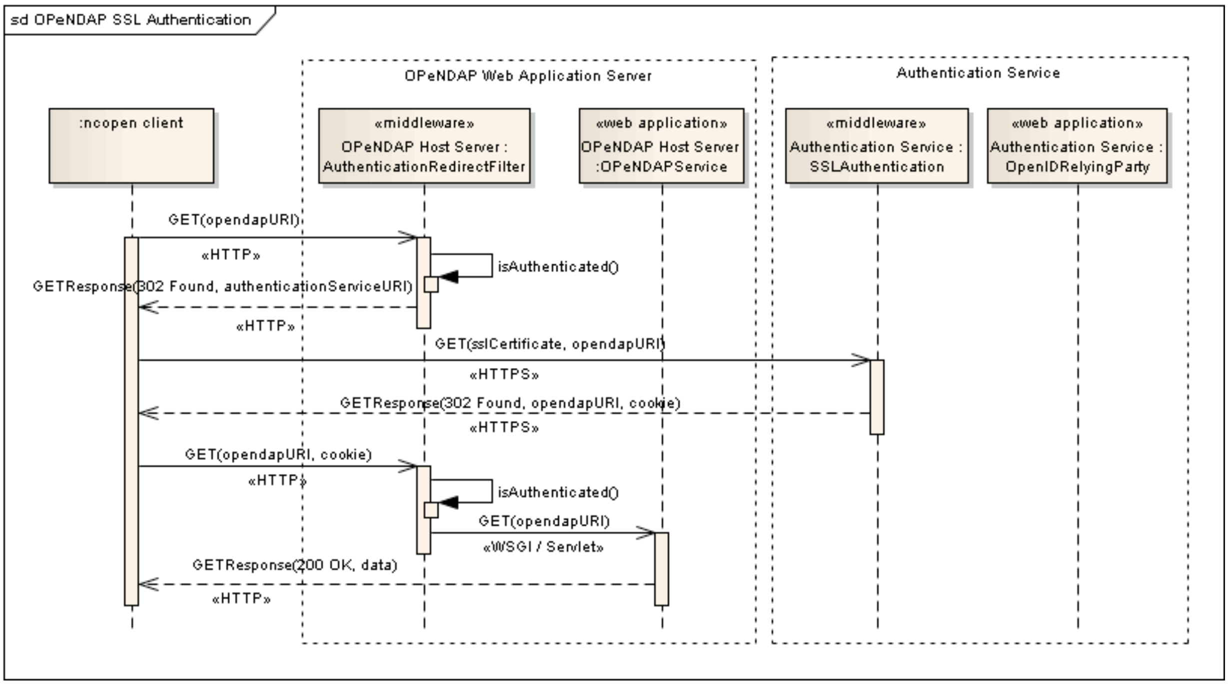

Figure 2: SSL Client Based Authentication with security filters (authorisation filters omitted for this illustration)

After the Authentication Filter, the request is intercepted by one or more authorization Filters. Typically, a chain will contain at least one SAML-based authorization filter that is responsible for issuing requests to the external Authorization Service. This may also enforce the authorisation decisions it receives or else delegate this to a separate dedicated enforcement filter.

Python and Java Implementations

A Python implementation has been made based on the NERC (UK Natural Environment Research Council) DataGrid Security system[3] developed at the BADC prior to ESGF. It employs the Python WSGI (Web Server Gateway Interface) specification which was used extensively to pilot many of the features of the filter based architecture described. A parallel Java filter based implementation has also been developed, which can be used to secure a generic web application that runs within a Java servlet container. This implementation is deployed widely at the ESGF Data Node sites.

5. Securing OPeNDAP Based Services

OPeNDAP is a data access framework widely used in the fields of oceanography and atmospheric science research, and was a key service to be supported by the ESGF security architecture. Data is served over a network interface, which abstracts the underlying data format from the client, and provides sub-setting functionality. For the Earth System Grid, OPeNDAP is a core protocol for serving data. The default Data Node configuration currently uses the THREDDS Data Server[21] (TDS) implementation of the OPeNDAP protocol, but other OPeNDAP servers will be supported soon (PyDAP is already deployed at BADC).

With the server side filter-based architecture as described in the previous section, it was possible to configure both TDS and PyDAP based OPeNDAP server implementations to support dual OpenID and SSL client based authentication mechanisms.

5.1. Extensions to NetCDF for ESGF Security-Aware Clients

Whilst the ability to apply this flexible approach to the server side security layer is important, the development of compatible client software is vital to the adoption of these services across a wide user base. The redirect-based pattern with PKI based credentials makes this solution suitable for simple HTTP based clients. Wget, a utility available on most UNIX based systems, can also be easily so configured.

Clearly though, for this solution to have significant adoption, the relevant changes would need to be integrated into OPeNDAP client libraries. The software libraries for NetCDF[13] were an obvious starting point. NetCDF is the standard format chosen for CMIP5 data and these are widely used as the basis for client tools in the climate science community. By inserting changes at the NetCDF level in the software stack, all these dependencies would collectively benefit. Unidata, the makers of the NetCDF software, were approached to look into such a possibility.

The C NetCDF API uses the *Curl* C libraries for HTTP calls. It was thus straightforward to make the necessary changes to enable SSL client settings. This results in client settings that are applied at the level of the user's `.dodsrc` file. In this way no change is made to the C API and so existing software that builds on the NetCDF libraries requires no change to source code to support ESGF based security, besides relinking with the latest version of the libraries. As of writing, the security extensions are available in a NetCDF beta2 release. This has been built with a number of different applications including *Ferret*[20], *ncview*[23] and the NetCDF Python bindings.

Work is underway to add support to the Java NetCDF client libraries and extensions to the PyDAP client libraries have enabled Pydap-based packages like CDX[19] to access ESGF hosted data. By instrumenting both NetCDF C/Java and PyDAP libraries, we are instantly enabling a large portion of the current earth science analysis toolkits with an access control layer.

6. Secured Data Access Walkthrough

At this point it is worth considering a walkthrough of a typical use case to illustrate how the individual components in the security architecture interact.

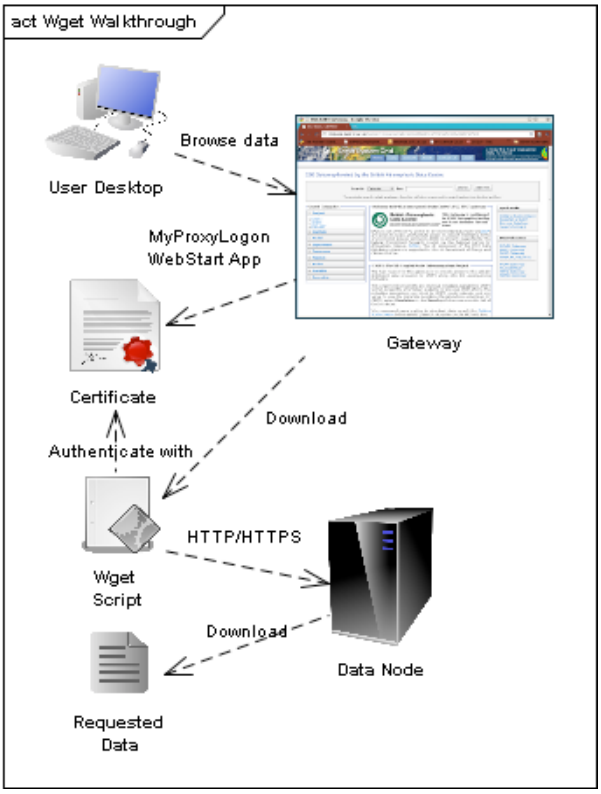

Figure 3: Secured Wget based data download

Imagine a user with a Web browser visiting a Gateway Node in the Federation and using its search facility to discover CMIP5 data for download. They find data hosted at the BADC's Data Node. Individual datasets may be downloaded directly via the browser. To facilitate multiple downloads, an option is provided to generate a script for the user to download and execute. This uses the wget utility to perform the HTTP based retrievals. To download secured datasets, the user needs to obtain short-term PKI credentials from their MyProxy online CA. The Gateway's user interface includes a Java MyProxyLogon[11] WebStart program that can be invoked for this purpose. The credentials are saved to a standard location on the user's file system from which the wget script can access them.

When the wget script is called, the various datasets are retrieved from the Data Nodes specified in the script download URLs. For any given request, the security filters fronting the data serving application authenticate the request based on the PKI credentials provided and check for authorisation by calling the respective Authorisation Services. For CMIP5 data, a given Authorisation Service will look up the corresponding authority for CMIP5 attribute registration: the PCMDI Attribute Service. This service asserts that the user has the correct entitlement and the Authorisation Service passes a decision back granting access.

7. Future Work and Related Developments

Although the initial deployment of the Earth System Grid Federation has been in the context of supporting CMIP5, many other applications are expected to be deployed using the same infrastructure. One such example is Live Access Server[20], to secure access to the data analysis and visualization capabilities it provides. Within both Europe and the U.S. there are major collaborative projects being built around ESGF, and future global collaborations will also exploit ESGF – and work on these has already begun with G8 funding.

Significantly, enabling PKI based authentication, opens up OPeNDAP based services to the Grid based security paradigm and in particular user delegation using proxy certificates[18]. A short NERC funded proof of concept project *MashMyData* is exploring how OPeNDAP services and an OGC Web Processing Service can be coupled together in a workflow. This use case leverages the ESGF security infrastructure with support for proxy certificates.

8. Conclusions

Modular design principles applied on a number of levels through the security architecture have resulted in a highly flexible solution applicable to the target domain whilst at the same time minimising its impact on existing services and tools.

The extensive use of existing standards in the Service Oriented Architecture has facilitated interoperability with Python and Java based implementations of services freely interchangeable. The filter based HTTP server side architecture has enabled the same access control solution to be applied over a range of applications. It has also made possible a flexible approach to access control configuration where any given application may be fronted with multiple authentication and authorisation schemes. This is demonstrated by dual OpenID and PKI based authentication support. The latter by exploiting characteristics inherent in

HTTP/HTTPS has minimised the entry point for client side tools to support it. This has meant that the user community can turn to simple freely available tools like wget to access secured data within ESGF. Finally, by applying security extensions to NetCDF, a software library used widely across the earth science community, all the dependent software packages and tools built on it are enabled with the security support.

9. Acknowledgements

We acknowledge the contributions of the various software development teams, their hard work and support in the full realisation of this architecture into a full implementation and deployment in an operational federation: the ESG-CET team, the wider ESGF development community and in particular: Stephen Pascoe (BADC), Neill Miller (ANL), Estanislao Gonzalez (MPIM, Hamburg), Gavin Bell, Bob Drach, Charles Doutriaux and the PCMDI development team; Nathan Wilhelmi, Eric Nienhouse and the development team at NCAR, Boulder, Colorado; Roland Schweitzer (NOAA/OAR). Finally, thanks also to Dean Williams and all the ESG-CET PIs.

The author also acknowledges the Software Sustainability Institute (UK) and NERC (UK) for their support with the NDG Security development precursor work.

This work was supported in part by the U.S. Dept. of Energy under Contract DE-AC02-06CH11357.

Also supported in part by the Jet Propulsion Laboratory, managed by the California Institute of Technology, under a contract with NASA.

10. References

[1] D N Williams, R Ananthakrishnan, D E Bernholdt, S Bharathi, D Brown, M Chen, A L Chervenak, L Cinquini, R Drach, I T Foster, P Fox, D Fraser, J Garcia, S Hankin, P Jones, D E Middleton, J Schwidder, R Schweitzer, R Schuler, A Shoshani, F Siebenlist, A Sim, W G Strand, M Su, N. Wilhelmi, The Earth System Grid: Enabling Access to Multi-Model Climate Simulation Data, in the Bulletin of the American Meteorological Society, February 2009.

[2] Siebenlist F, R. Ananthakrishnan, D. E. Bernholdt, L. Cinquini, I. T. Foster, D. E. Middleton, N. Miller, D. N. Williams, Earth System Grid Authentication Infrastructure: Integrating Local Authentication, OpenID and PKI, TeraGrid 2009, June 2009

[3] Kershaw, Philip, Jon Blower, Bryan Lawrence, Practical Access Control using NDG Security, e-Science All Hands Meeting, September 2007

[4] Sinnott, R.O., Chadwick, D.W., Koetsier, J., Otenko, O., Watt J. and Nguyen, T.A. (2006) Supporting decentralized, security focused dynamic virtual organizations across the grid, Proceedings of the Second IEEE International Conference on e-Science and Grid Computing 2006 (e-Science '06), December 2006, Amsterdam, The Netherlands.

[5] Barton T, Basney J, FreemanT, Scavo T, Siebenlist F, Welch V, Ananthakrishnan R, Baker B, Goode M, Keahey K, Identity Federation and Attribute-based Authorization through the Globus Toolkit, Shibboleth, GridShib, and MyProxy, 5th Annual PKI R&D Workshop, 2006

[6] Kiczales, G., Lamping, J., Mendhekar, A., Maeda, C., Videira Lopes, C., Loingtier, J.-M., and Irwin, J, Aspect-Oriented Programming, Proceedings of ECOOP 1997.

[7] Representational State Transfer (REST), Fielding R.T., Architectural Styles and the Design of Network-based Software Architectures, http://www.ics.uci.edu/~fielding/pubs/dissertation/top.htm

[8] OASIS Security Services (SAML) TC, http://www.oasis-open.org/committees/tc_home.php?wg_abbrev=security

[9] OpenID, http://openid.net

[10] Yadis, http://yadis.org

[11] MyProxy, http://grid.ncsa.uiuc.edu/myproxy/

[12] OPeNDAP, http://www.opendap.org

[13] NetCDF (Network Common Data Form), http://www.unidata.ucar.edu/software/netcdf/

[14] GridFTP, http://dev.globus.org/wiki/GridFTP

[15] Web Server Gateway Interface, http://wsgi.org/wsgi

[16] PyDAP, http://pydap.org

[17] X.509 Certificate, http://tools.ietf.org/html/rfc5280

[18] Internet X.509 Public Key Infrastructure (PKI) Proxy Certificate Profile, http://www.ietf.org/rfc/rfc3820.txt

[19] CDX, http://cdx.jpl.nasa.gov/

[20] Ferret, http://ferret.wrc.noaa.gov/

[21] THREDDS Data Server, http://www.unidata.ucar.edu/projects/THREDDS/tech/TDS.html

[22] D N Williams and K E Taylor (PCMDI), B N Lawrence (BADC), and M Lautenschlager (WDCC), How will CMIP5 Model Output be Archived and Served? http://cmip-pcmdi.llnl.gov/cmip5/submit.html?submenuheader=2

[23] ncview, http://meteora.ucsd.edu/~pierce/ncview_home_page.html